

upon determining that the stream in the SCC system is different than the stream in the database, providing a user a choice to select one of the stream in the SCC system and the stream in the database; and

upon the user selecting the stream in the SCC system, saving the stream in the SCC system as the stream in the database.

A5 *Sub E10*

16. The computer-readable medium of claim 15, the method further comprising prior to comparing:

retrieving the stream of the item from the source code control (SCC) system;

and

retrieving the stream of the item from the database.

17. The computer-readable medium of claim 15, the method further comprising:

retrieving a stream of the item from a database; and

saving the stream of the item in the database as the stream in the source code control (SCC) system.

REMARKS

Claims 1-18 were rejected. Claims 1, 4-8, 10-13, and 15-17 have been amended. Claims 1-18 are now pending in this application.

35 USC §112 Rejection of the Claims

Claims 1-18 were rejected under 35 USC §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicants regard as the invention. In particular, the Office Action indicates that the acronym “SCC” is not clear. The claims have been amended to clarify that the acronym “SCC” refers to “source code control.” Accordingly, these changes are for clarification purposes, and are unrelated to patentability.

35 USC §102(e) Rejection of the Claims

Claims 1-18 were rejected under 35 USC § 102(e) as being anticipated by House et al. (U.S. Patent No. 6,145,119) (“House”). According to the Office action, House allegedly discloses: a program that manipulates an item (column 2, lines 4-8), a database having the item (column 3, lines 44-50), a source code control system to store versions of the item (column 6, lines 17-34 and column 7, lines 52-67) and a mechanism to check in and check out the item (column 5, lines 10-27).

However, because House does not teach or suggest using a source code control (SCC) system to store versions of an item, and because House does not teach or suggest a mechanism to check in and check out the item, as recited in claim 1 of the present invention, Applicants respectfully traverse the rejection.

House discloses a method for providing a programming development environment that supports the accessing of database software across an Internet or Intranet via Web-based networks (*House* – column 2, lines 49-55). In so doing, House provides a data structure that allows all elements and associations necessary to build the components of a project, such as source, objects, and executables to be located in a single file (*House* – column 2, lines 57-60).

Such a data structure makes group development easier and facilitates file sharing (*House* – column 2, lines 61-63). However, House does not teach or suggest the elements disclosed in claim 1 of the present invention. For example, House does not teach or suggest providing a source code control (SCC) system that stores versions of an item. Also, House does not teach or suggest providing a mechanism to check in and check out the item, as with the present invention. This is to be expected, because as its title indicates, House is directed toward using a novel data structure to facilitate a programming development environment conducive to Intranet and Internet applications. The present invention, on the other hand, is directed toward, *inter alia*, checking in and out an item that is stored in a SCC system.

The Office action first asserts that the present invention's claim of "a SCC system to store versions of the item" allegedly is taught by the APP file 454. House describes "the APP file 454 as the persistence mechanism for the Object Model that all components of the present invention are built upon" (*House* – column 6, lines 17-20). House notes that although the APP file is not "humanly readable," its data structure "comprises two major sections, a Runtime Section 550 and an Other Data section 552. The Runtime Section 550 contains all of the data needed primarily at 'runtime,' including executable programming logic. . . . The Other Data Section 552, . . . contains all other project related data, including project environment data (such as source code, design time properties, other project development data, etc.). By segmenting the APP file 454 into these two sections, the speed of execution is enhanced, because reading in the Other Data Section 552 can be skipped when executing an application" (*House* – column 6, lines 21-34).

This description of House's APP file is to be distinguished from the present invention's disclosure of a SCC system used to store versions of an item. Contrary to the present invention,

House does not teach or suggest using a system to control source code by storing a version of an item to be checked in and out by a mechanism. Instead, House describes a data structure that speeds execution by segmenting data into two sections (*i.e.*, Runtime Section 550 and Other Data Section 552).

The office action further asserts that the present invention's claim of "a mechanism to check in and check out the item" allegedly is taught by Houses's description with reference to its Figure 4. Although the present invention is not so limited, in one embodiment, "checking out" an item refers to the process of tagging an item as "in use" by one user, such that another user cannot edit or modify the item until it has been checked back in (*Specification* - page 14, lines 9-14).

Although not specifically cited, it appears that the Office action alleges that the mechanism of the present invention is taught by "[a] development computer 400 [that] executes a Rapid Application Development (RAD) tool comprised of a number of different computer programs or modules, including a graphical user interface (GUI) 402, project manager 404 and associated builder 406, form editor 408 for constructing HTML forms 410" (*House* - column 5, lines 11-16). However, no where does House teach that such a development computer permits an item to be checked in and out. On the contrary, by facilitating "file sharing," House teaches away from the present invention's claimed checking in and out of an item.

Therefore, House does not anticipate the invention as recited in claim 1. For the same reasons discussed above, claims 2-9, which depend from claim 1, also are not anticipated by House. Accordingly, Applicants respectfully request that the rejection of claim 1 and claims 2-9 under 35 USC §102(e) be withdrawn.

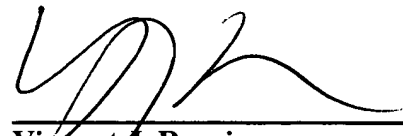
For the same reasons expressed above with respect to claims 1-9, Applicants respectfully request withdraw of the rejection under 35 USC §102(e) of independent claim 10 and its dependent claims 11-14, and independent claim 15 and its dependent claims 16-18.

CONCLUSION

In view of the foregoing amendments and remarks, Applicants respectfully submit that the present application is in condition for allowance. Reconsideration of the application and an early Notice of Allowance are respectfully requested. In the event that the Examiner cannot allow the present application for any reason, the Examiner is encouraged to contact Applicants' attorney at (215-564-8946).

Date: 4/18, 2001

Respectfully submitted,



Vincent J. Roccia
Registration No. 43,887

WOODCOCK WASHBURN KURTZ
MACKIEWICZ & NORRIS LLP
One Liberty Place - 46th Floor
Philadelphia, PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439

Marked up version of the first full paragraph appearing at page 27, which spanned lines 1 through 9 in the specification as originally filed, and which is replaced herein.

The invention provides [Database] database item versioning, such as items like stored procedures found in databases such as Standard Query Language (SQL) databases. In one embodiment, the invention includes a program such as an editor, a database, a source code control and a mechanism. An editor program provides for editing a stored procedure. The [databases] database includes the stored procedure. The [SCC] source code control system stores versions of the stored procedure. The mechanism checks in and checks out the stored procedure to provide for effective versioning [thereof] of the stored procedure.